Assignment #8

CSCI 581, Spring 2022

Jayaa Emekar

Finding clusters in the Iris dataset

You will be treating the Iris dataset, available from Scikit-learn, as unlabeled data to perform clustering via the `KMeans` estimator.

Overview

Scikit-learn's Iris dataset is labeled data that we have used in supervised learning. To use it as unlabeled data for unsupervised learning, you will need to drop the last column from the training set before running the `fit()` method on an instance of the `KMeans` estimator.

Instructions

1. Since we already know there are three varieties of Irises embodied in the Iris dataset, use the whole unlabeled dataset with $k = 3$ on an instance of the `KMeans` estimator. Evaluate the performance of the model using as many metrics as you deem appropriate.
2. Confirm if $k = 3$ is the optimum value for k on the unlabeled version of the Iris dataset. Use the Silhouette Coefficient as a metric similar to the examples shown in our PML 10 jupyter notebook.
3. Summarize all your findings and present your conclusions regarding the use of this algorithm on the Iris dataset.

Required components of your submission

Your *Google Colab* Jupyter notebook must include:

1. all pertinent *exploratory data analysis* (EDA) code, visualizations, and justifications (you can reuse, perhaps with minimal modification, the work you did in your earlier Assignments);
2. explanations/justifications for all model selection decisions;
3. all pertinent model diagnostics, including metrics and visualizations; and
4. your summary and conclusions pertaining to how the two models compare against each other.

Be sure to check out or review the *Assignments/Projects* section of our *Blackboard* course page for details regarding expectations, requirements, and the *Jupyter Notebook Rubric* that

will be used to evaluate Jupyter notebook submissions.

Solution

All required imports for solution

In [ ]:
```
''' all required imports '''
from sklearn.datasets import load_iris
from sklearn.cluster import KMeans
import seaborn as sns; sns.set_theme(color_codes=True)
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import itertools
```

load dataset

Load the iris dataset

In [ ]:
```
iris = sns.load_dataset("iris")
```

Expolratory Data Analysis and preprocessing

Lets have initial overview of dataset

Print dataset info

In [ ]:
```
# initial overview of dataset
print("Dataset info:\n", iris.info(), '\n\n')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
Dataset info:
 None
```

Print dataset description

In [ ]:
```
print("Dataset describtion:\n", iris.describe(), '\n\n')
```

```
Dataset describtion:
        sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
```

Print first five rows from the dataset

In [ ]:
```
print("Dataset first five rows:\n", iris.head(), '\n\n')
```

```
Dataset first five rows:
    sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

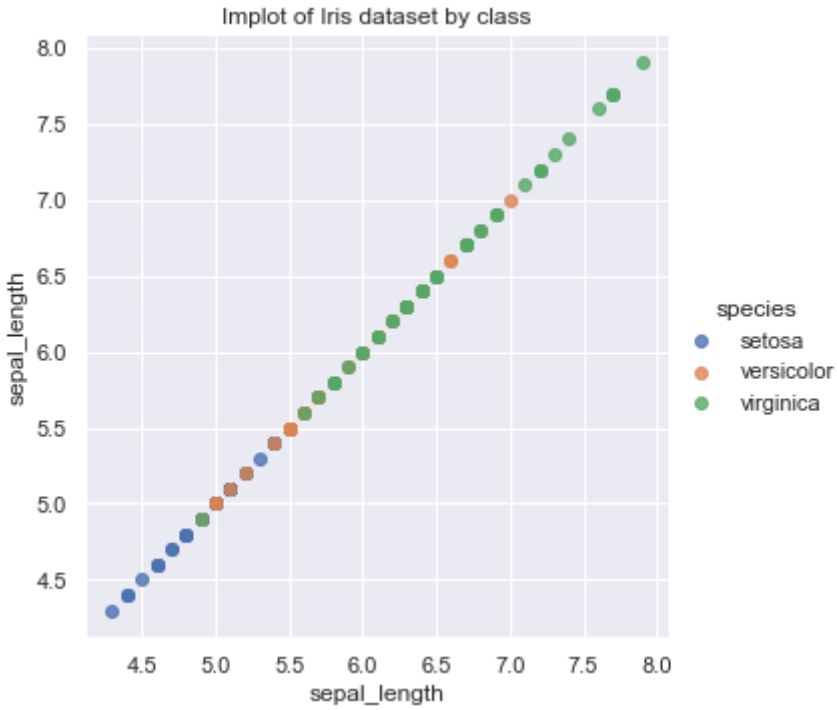
Print null values count from the dataset

In [ ]:
```
# Print null values count from the dataset
print("Dataset null values count:\n", iris.isnull().sum(), '\n\n')
```

```
Dataset null values count:
 sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Lets visualize the correlation with the target -- just to get an idea

In [ ]:
```
# visualize the correlation with the target -- just to get an idea
features = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
comboFeatures = list(itertools.product(features, repeat=2))
sns.set(font_scale=1)
for combo in comboFeatures:
    sns.lmplot(x=combo[0], y=combo[1], data=iris,
               hue="species", height=5, aspect=1, fit_reg=False)
    plt.title("Implot of Iris dataset by class")
    plt.show()
    plt.clf()
```



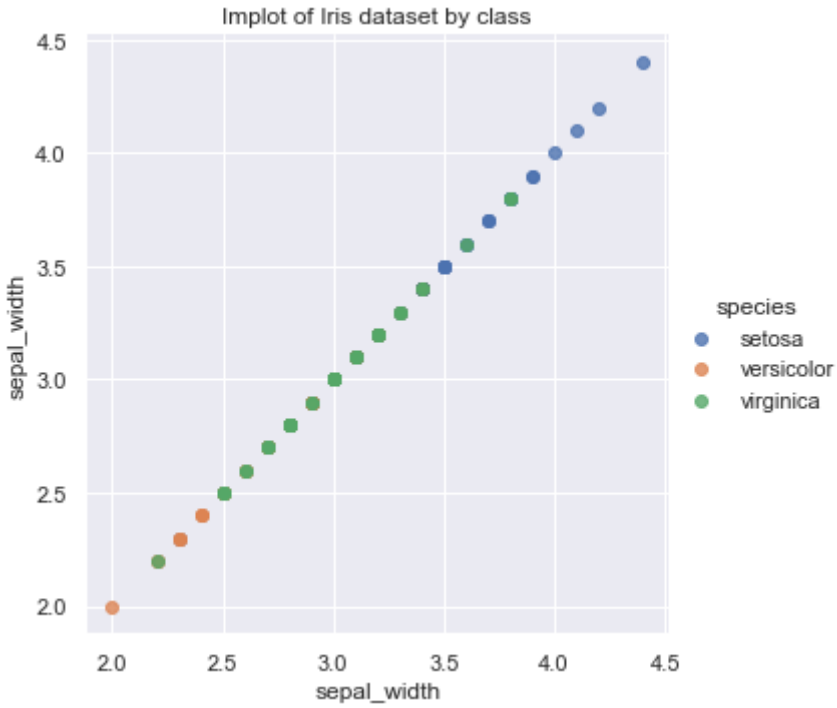
<Figure size 432x288 with 0 Axes>

<Figure size 432x288 with 0 Axes>

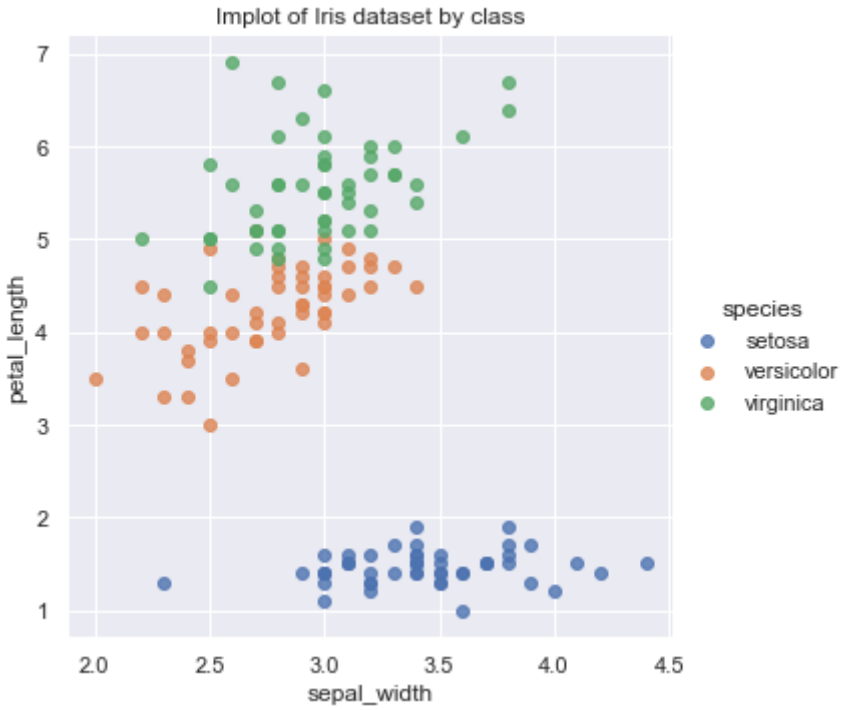

```
<Figure size 432x288 with 0 Axes>
```


```
<Figure size 432x288 with 0 Axes>
```

<Figure size 432x288 with 0 Axes>

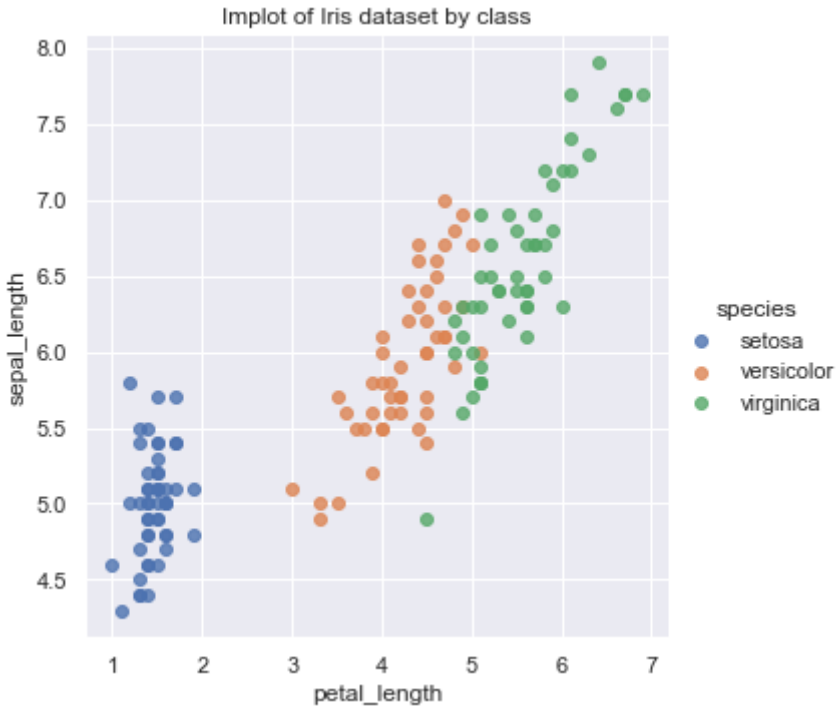


<Figure size 432x288 with 0 Axes>

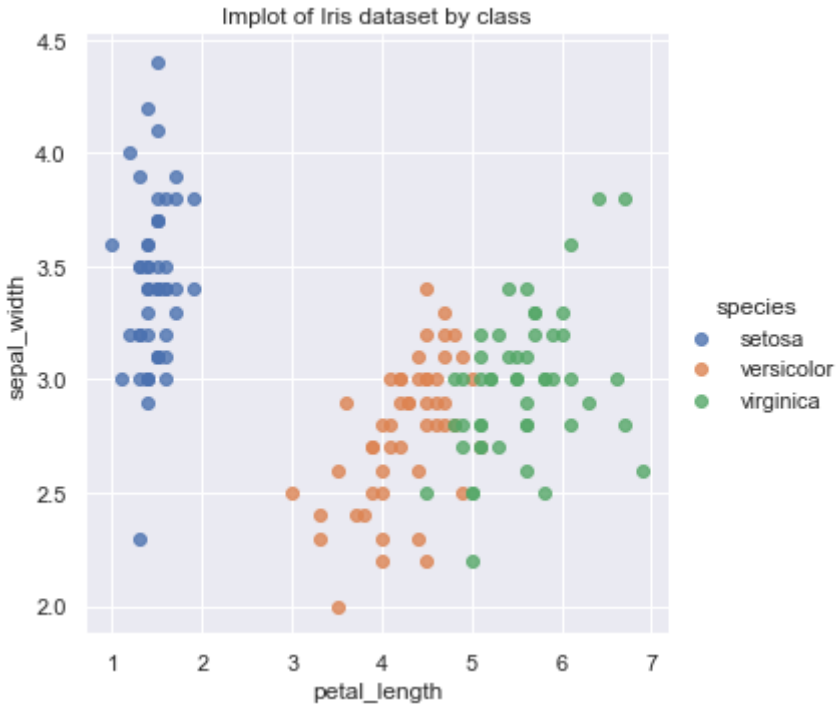


<Figure size 432x288 with 0 Axes>

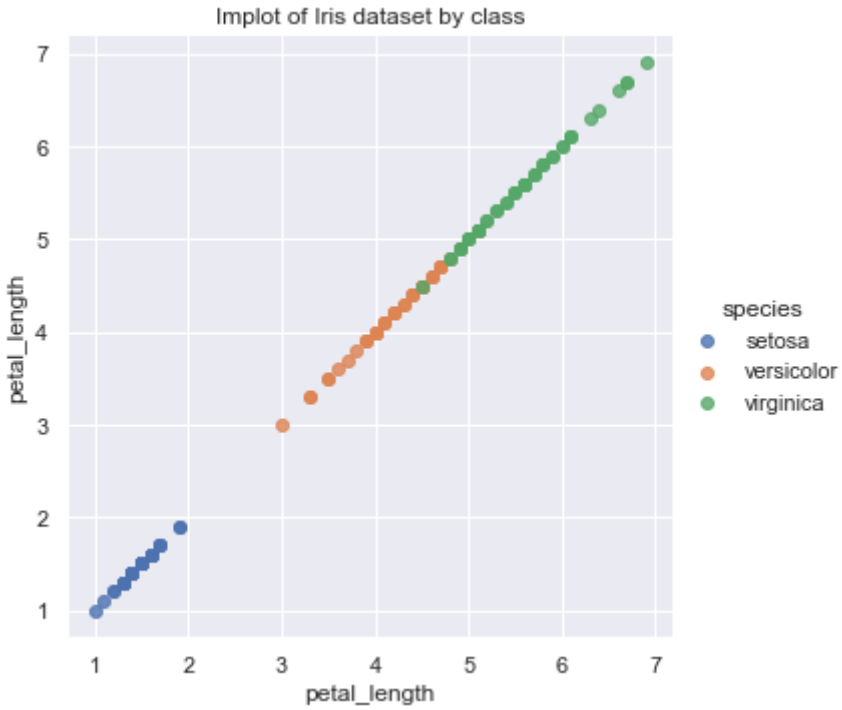
<Figure size 432x288 with 0 Axes>



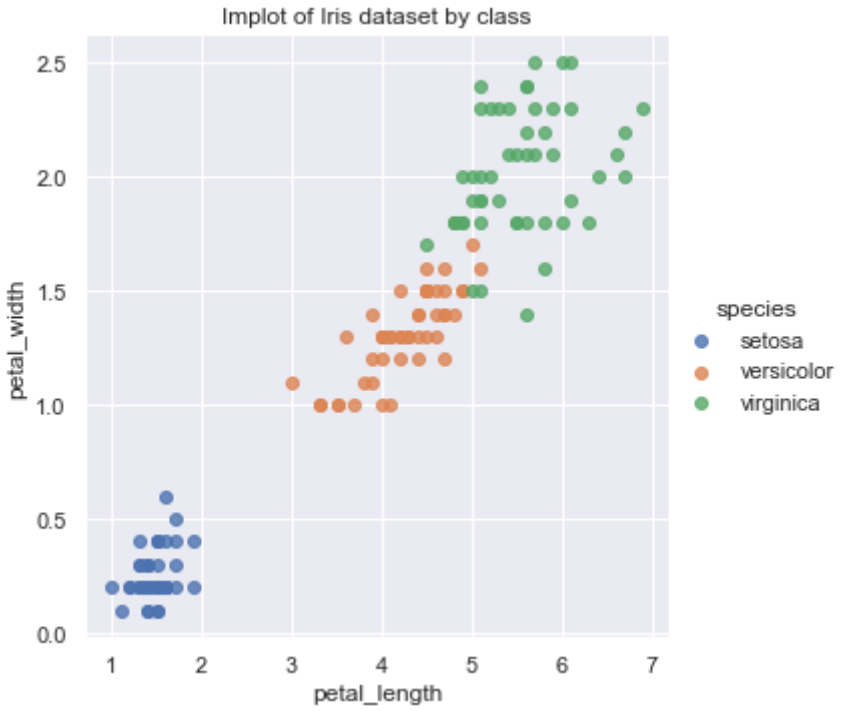
<Figure size 432x288 with 0 Axes>



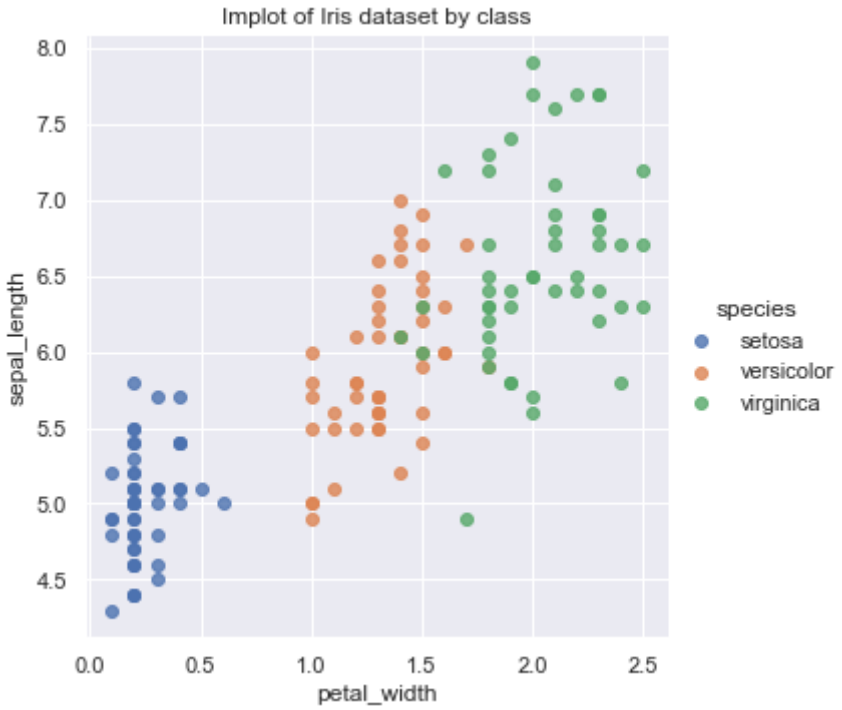
<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>

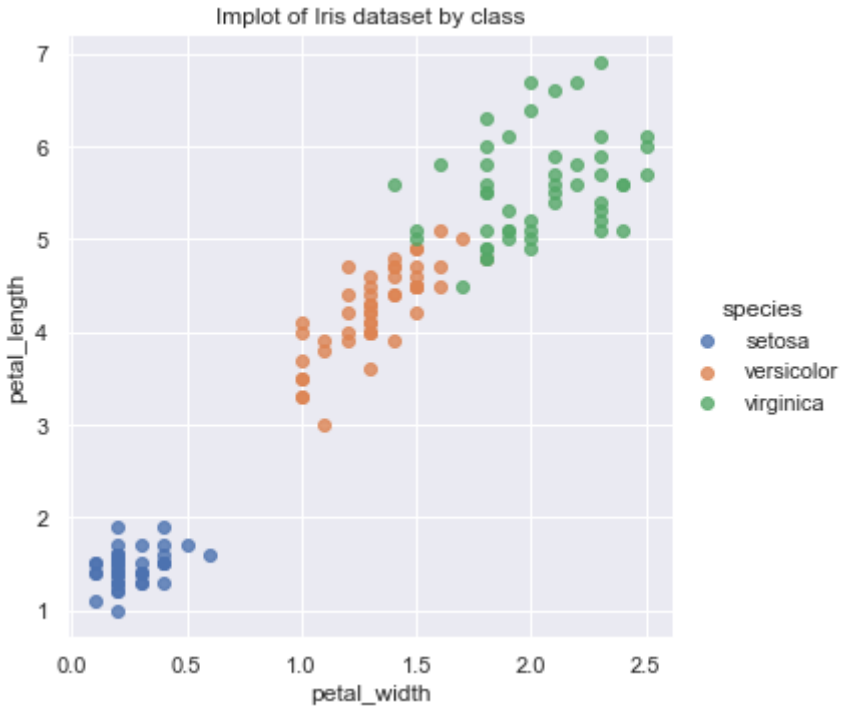


<Figure size 432x288 with 0 Axes>

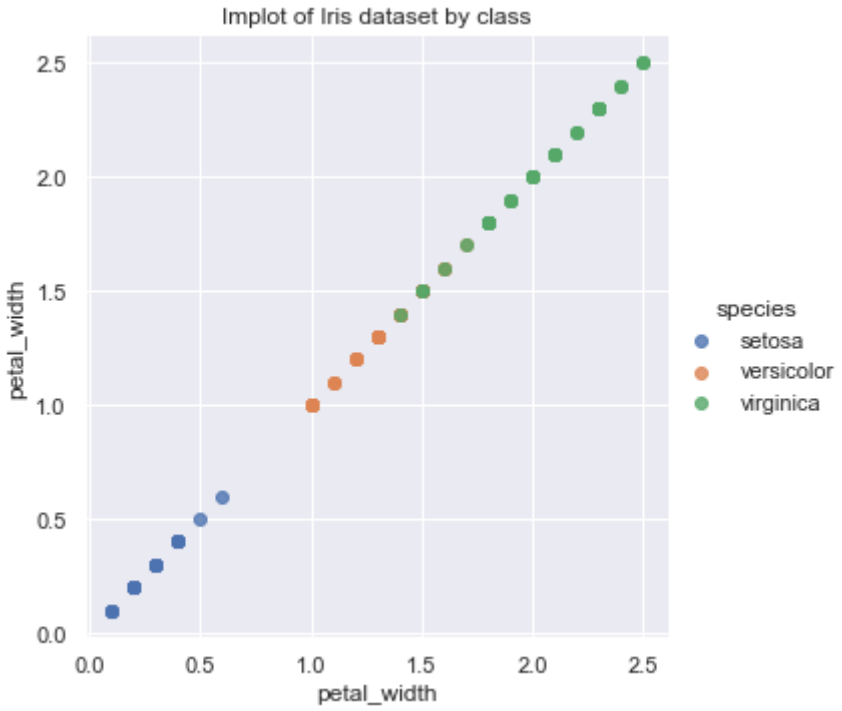


<Figure size 432x288 with 0 Axes>

<Figure size 432x288 with 0 Axes>




```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

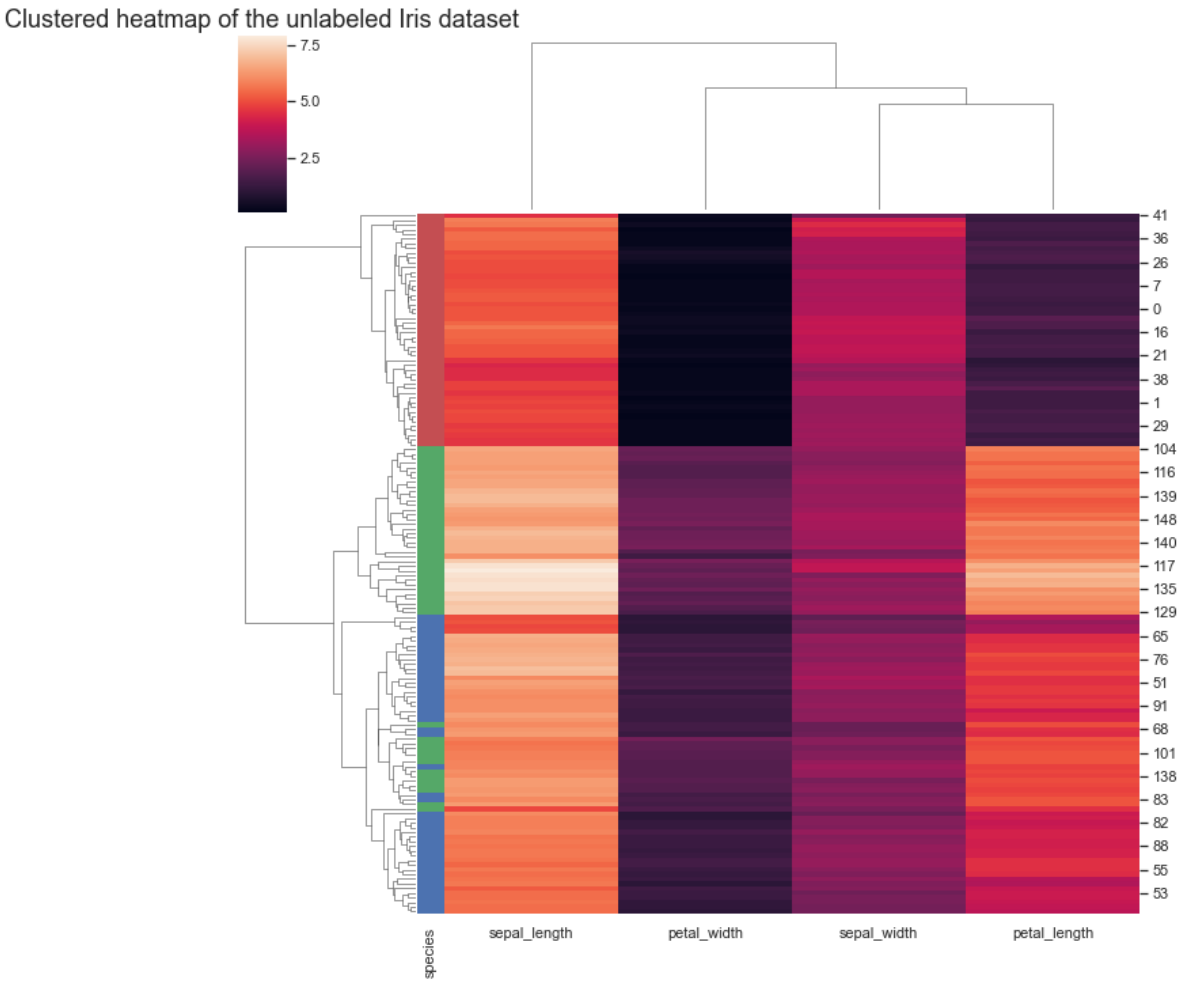
In above plot we have seen the corralation between sepal_width, sepal_length, petal_width and petal_length for the different species now we got fair idea about the corralation between the species

Lets visualize the clustering of the dataset features and target

In [ ]:
```
# visualize the clustering of the dataset features and target
iris_copy = iris.copy()
species = iris_copy.pop("species")   # drop the target to use the dataset as unlabe
print(f"the dataset has {len(species.unique().tolist())} classes: {species.unique()
lut = dict(zip(species.unique(), "rbg"))
row_colors = species.map(lut)
g = sns.clustermap(iris_copy, row_colors=row_colors)
sns.set(font_scale=1.5)
```

```
plt.title("Clustered heatmap of the unlabeled Iris dataset")
plt.show()
plt.clf()
```

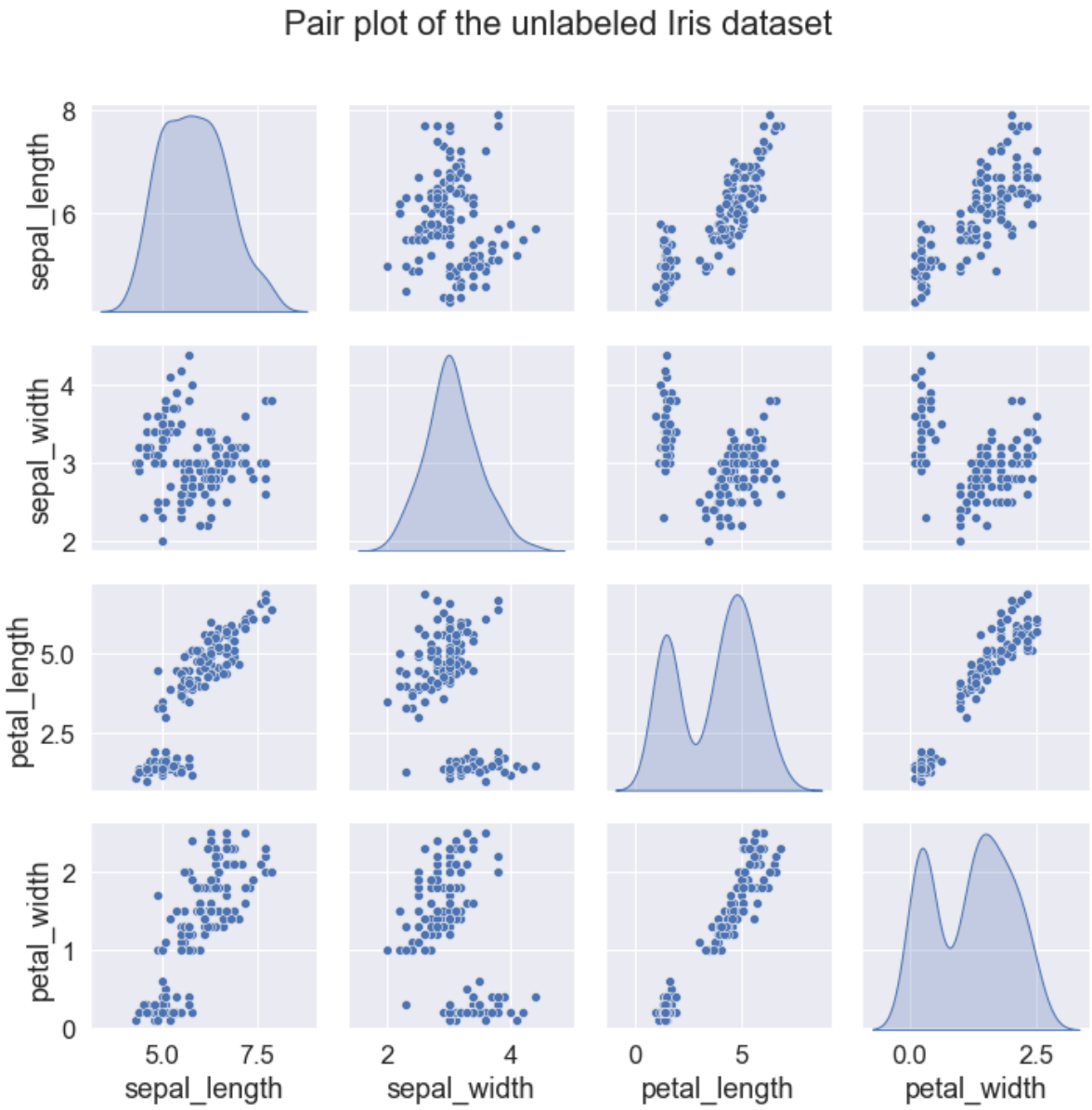
```
the dataset has 3 classes: ['setosa' 'versicolor' 'virginica']
```




```
<Figure size 432x288 with 0 Axes>
```

Lets visualize the correlation of the features to the target

In [ ]:
```
# visualize the correlation of the features to the target
pp = sns.pairplot(iris_copy, diag_kind='kde')
pp.fig.suptitle("Pair plot of the unlabeled Iris dataset", y=1)
plt.tight_layout()
plt.show()
plt.clf()
```

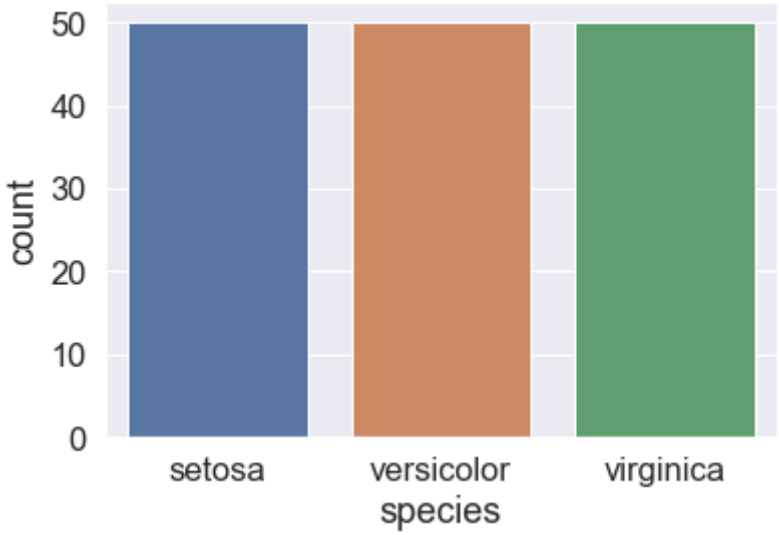



```
<Figure size 432x288 with 0 Axes>
```

Lets inspect any bias in the dataset

In [ ]:
```
# inspect any bias in the dataset
sns.countplot(iris.species)
plt.show()
plt.clf()
```

```
C:\Users\shrih\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_d
ecorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. F
rom version 0.12, the only valid positional argument will be `data`, and passing o
ther arguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```




```
<Figure size 432x288 with 0 Axes>
```

Observation

From the above EDA analysis, we can infer that –

1. Species Setosa has smaller sepal lengths but larger sepal widths.
2. Versicolor Species lies in the middle of the other two species in terms of sepal length and width
3. Species Virginica has larger sepal lengths but smaller sepal widths.
4. Species Setosa has smaller petal lengths and widths.
5. Versicolor Species lies in the middle of the other two species in terms of petal length and width
6. Species Virginica has the largest of petal lengths and widths.
7. In heatmap, We can see many types of relationships from this plot such as the species Seotsa has the smallest of petals widths and lengths. It also has the smallest sepal length but larger sepal widths. Such information can be gathered about any other species.
8. The highest frequency of the sepal length is between 30 and 35 which is between 5.5 and 6
9. The highest frequency of the sepal Width is around 70 which is between 3.0 and 3.5
10. The highest frequency of the petal length is around 50 which is between 1 and 2
11. The highest frequency of the petal width is between 40 and 50 which is between 0.0 and 0.5
12. Petal width and petal length have high correlations.
13. Petal length and sepal width have good correlations.
14. Petal Width and Sepal length have good correlations.
15. Species Setosa has the smallest features and less distributed with some outliers.
16. Species Versicolor has the average features.
17. Species Virginica has the highest features
18. In the case of Sepal Length, there is a huge amount of overlapping.
19. In the case of Sepal Width also, there is a huge amount of overlapping.
20. In the case of Petal Length, there is a very little amount of overlapping.
21. In the case of Petal Width also, there is a very little amount of overlapping.

Building Model

Lets split the dataset into features and target values

In [ ]:
```
''' split the dataset '''

X = iris.iloc[:,:-1].values # features on (i.e., unlabeled dataset)
y = iris.iloc[:,-1].values # target values
```

Lets start building model for dataset

We used number of clusters as 3

In [ ]:
```
kmeans = KMeans(n_clusters=3, random_state=23)
y_pred = kmeans.fit_predict(X)
```

Lets print the predictions

In [ ]:
```
print(y_pred)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 2
 1 2]
```

print kmeans lable

In [ ]:
```
print(kmeans.labels_)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 2
 1 2]
```

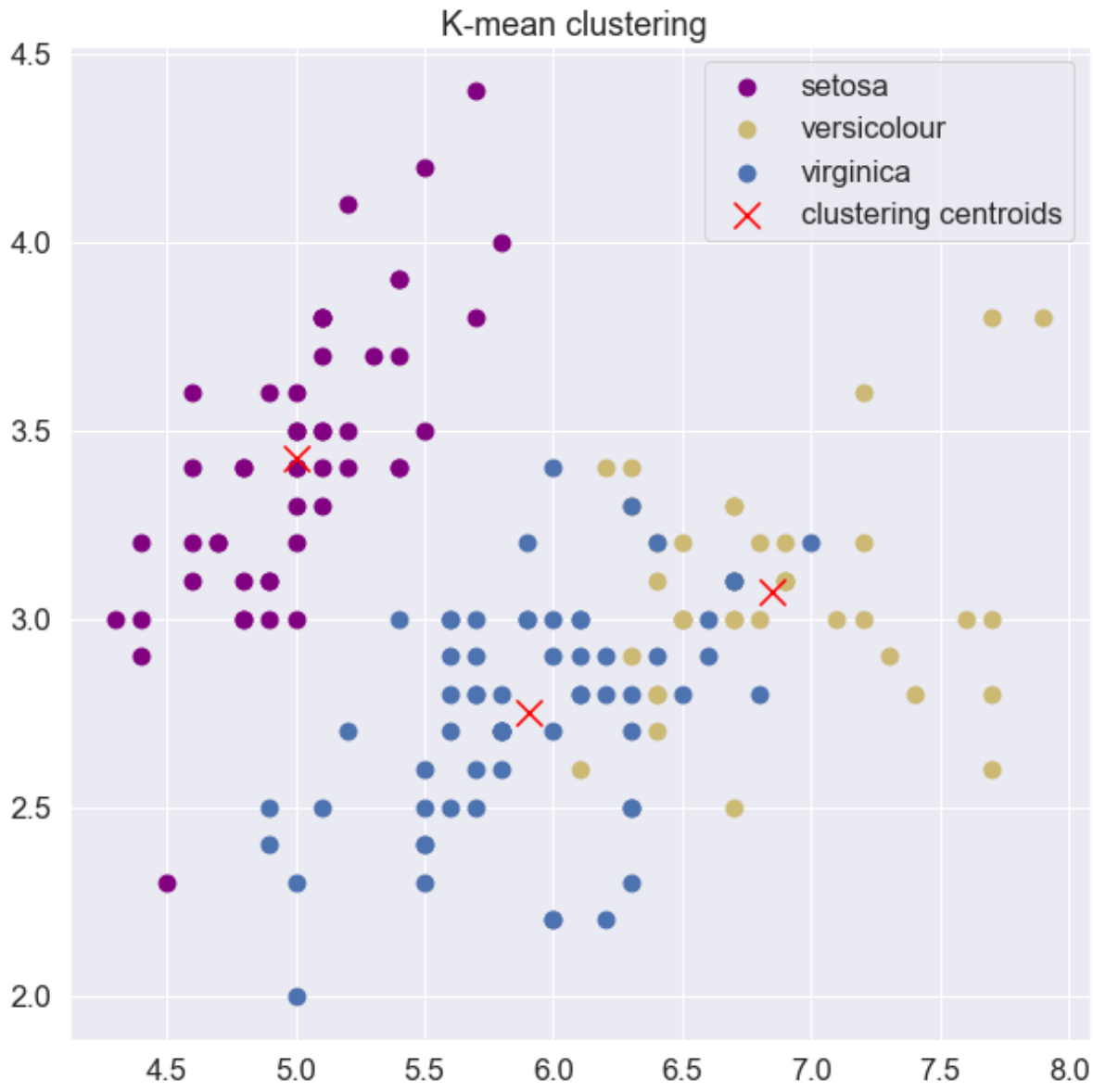
print kmeans cluster centers

In [ ]:
```
print(kmeans.cluster_centers_)
```

```
[[5.006      3.428      1.462      0.246     ]
 [6.85       3.07368421 5.74210526 2.07105263]
 [5.9016129  2.7483871  4.39354839 1.43387097]]
```

Lets Visualising the clusters - On the first two columns

In [ ]:
```
plt.figure(figsize=(10,10))
plt.scatter(X[y_pred == 0, 0], X[y_pred == 0, 1],
            s=80, color='purple', label='setosa')
plt.scatter(X[y_pred == 1, 0], X[y_pred == 1, 1],
            s=80, color='y', label='versicolour')
plt.scatter(X[y_pred == 2, 0], X[y_pred == 2, 1],
            s=80, color='b', label='virginica')   #Visualising the clusters - On t
plt.scatter(kmeans.cluster_centers_[:, 0],
            kmeans.cluster_centers_[:, 1],
            s=200, marker='x', color='red', label='clustering centroids')   #plott
plt.title("K-mean clustering ")
plt.legend()
plt.show()
```



Lets evaluate model performance using Silhouette Coefficient

In [ ]:
```
''' evaluate model performance using Silhouette Coefficient '''

from sklearn import metrics

silhouette_samples = metrics.silhouette_samples(X, kmeans.labels_)
print(silhouette_samples)
```

```
[0.85295506 0.81549476 0.8293151  0.80501395 0.8493016  0.74828037
 0.82165093 0.85390505 0.75215011 0.825294   0.80310303 0.83591262
 0.81056389 0.74615046 0.70259371 0.64377156 0.77568391 0.85101831
 0.70685782 0.82030124 0.78418399 0.82590584 0.79297218 0.7941134
 0.77503635 0.79865509 0.83346695 0.84201773 0.84364429 0.81784646
 0.81518962 0.79899235 0.76272528 0.72224615 0.82877171 0.83224831
 0.79415322 0.84188954 0.76856774 0.85033231 0.84941579 0.63900017
 0.78657771 0.80023815 0.74698726 0.80977534 0.81340268 0.81902059
 0.8182324  0.85209835 0.02672203 0.38118643 0.05340075 0.59294381
 0.36885321 0.59221025 0.28232583 0.26525405 0.34419223 0.57829491
 0.37478707 0.58710354 0.55107857 0.48216686 0.56310057 0.32459291
 0.55751057 0.61072967 0.46149897 0.6115753  0.32909528 0.58968904
 0.31046301 0.49424779 0.5000461  0.38548959 0.12629433 0.11798213
 0.55293611 0.5069822  0.59466094 0.5607585  0.61972579 0.26087292
 0.54077013 0.41598629 0.16655431 0.48935747 0.60716023 0.61436443
 0.59560929 0.50352722 0.62444848 0.29362234 0.62754454 0.60657448
 0.62205599 0.55780204 0.14131742 0.63064081 0.49927538 0.23225278
 0.61193633 0.36075942 0.5577792  0.54384277 0.46682151 0.55917348
 0.44076207 0.56152256 0.26062588 0.22965423 0.55509948 0.28503067
 0.02635881 0.39825264 0.42110831 0.49486598 0.48341063 0.32868889
 0.6070348  0.33355947 0.51237366 0.20297372 0.580154   0.57818326
 0.30904249 0.25226992 0.45434264 0.51608826 0.56017398 0.48442397
 0.46255248 0.13900039 0.05328614 0.55186784 0.45549975 0.3887791
 0.35124673 0.53444618 0.5702338  0.41025549 0.23225278 0.61324746
 0.5670778  0.42513648 0.10417086 0.31493016 0.35245379 0.18544229]
```

Average of Silhouette Coefficients for k =3 can we obtained by

In [ ]:
```
print("Average of Silhouette Coefficients for k =", 3)
print("============================================")
print("Silhouette mean:", silhouette_samples.mean())
```

```
Average of Silhouette Coefficients for k = 3
============================================
Silhouette mean: 0.5528190123564102
```

finding the optimal K value

In [ ]:
```
''' finding the optimal K '''

silhouette_avgs = []
min_k = 2

#---try k from 2 to maximum number of labels---
for k in range(min_k, len(X)):
    kmean = KMeans(n_clusters=k, random_state=23).fit(X)
    score = metrics.silhouette_score(X, kmean.labels_)
    print("Silhouette Coefficients for k =", k, "is", score)
    silhouette_avgs.append(score)
```

```
Silhouette Coefficients for k = 2 is 0.6810461692117465
Silhouette Coefficients for k = 3 is 0.5528190123564102
Silhouette Coefficients for k = 4 is 0.49805050499728815
Silhouette Coefficients for k = 5 is 0.48874888709310654
Silhouette Coefficients for k = 6 is 0.36483400396700366
Silhouette Coefficients for k = 7 is 0.35817224727219793
Silhouette Coefficients for k = 8 is 0.3448276330056002
Silhouette Coefficients for k = 9 is 0.3375652296778145
Silhouette Coefficients for k = 10 is 0.30209683941954024
Silhouette Coefficients for k = 11 is 0.2961219167135574
Silhouette Coefficients for k = 12 is 0.3091096992308015
Silhouette Coefficients for k = 13 is 0.2648401041878877
Silhouette Coefficients for k = 14 is 0.2876849784879395
Silhouette Coefficients for k = 15 is 0.3024146377875319
Silhouette Coefficients for k = 16 is 0.2600089401558184
Silhouette Coefficients for k = 17 is 0.26945907971710686
Silhouette Coefficients for k = 18 is 0.2703684401364715
Silhouette Coefficients for k = 19 is 0.27239617379065045
Silhouette Coefficients for k = 20 is 0.28553110471422316
Silhouette Coefficients for k = 21 is 0.28153517861115307
Silhouette Coefficients for k = 22 is 0.27765547681430586
Silhouette Coefficients for k = 23 is 0.2762684183260351
Silhouette Coefficients for k = 24 is 0.2677288491326929
Silhouette Coefficients for k = 25 is 0.2622672954738558
Silhouette Coefficients for k = 26 is 0.25981617872474994
Silhouette Coefficients for k = 27 is 0.2554889595887313
Silhouette Coefficients for k = 28 is 0.26930809760461627
Silhouette Coefficients for k = 29 is 0.26864766108578675
Silhouette Coefficients for k = 30 is 0.2537881195582788
Silhouette Coefficients for k = 31 is 0.2694384395652744
Silhouette Coefficients for k = 32 is 0.2530982955465582
Silhouette Coefficients for k = 33 is 0.2578138618220197
Silhouette Coefficients for k = 34 is 0.2656939260671239
Silhouette Coefficients for k = 35 is 0.2658106138176077
Silhouette Coefficients for k = 36 is 0.2646974556225076
Silhouette Coefficients for k = 37 is 0.2734534496821079
Silhouette Coefficients for k = 38 is 0.2742550543828211
Silhouette Coefficients for k = 39 is 0.2506369524864321
Silhouette Coefficients for k = 40 is 0.2596403577254779
Silhouette Coefficients for k = 41 is 0.2737992225770712
Silhouette Coefficients for k = 42 is 0.23456630220372962
Silhouette Coefficients for k = 43 is 0.26625691333790036
Silhouette Coefficients for k = 44 is 0.2610825466544045
Silhouette Coefficients for k = 45 is 0.2825023403662287
Silhouette Coefficients for k = 46 is 0.26029131410714623
Silhouette Coefficients for k = 47 is 0.27500675909731614
Silhouette Coefficients for k = 48 is 0.2646863442975752
Silhouette Coefficients for k = 49 is 0.2750073334673118
Silhouette Coefficients for k = 50 is 0.25909784675730585
Silhouette Coefficients for k = 51 is 0.2569300557279579
Silhouette Coefficients for k = 52 is 0.266208175017224
Silhouette Coefficients for k = 53 is 0.28184608965258123
Silhouette Coefficients for k = 54 is 0.2836483655557778
Silhouette Coefficients for k = 55 is 0.2604255536673985
Silhouette Coefficients for k = 56 is 0.2786058096970785
Silhouette Coefficients for k = 57 is 0.2840363689654772
Silhouette Coefficients for k = 58 is 0.26841951546742054
Silhouette Coefficients for k = 59 is 0.2630293969774437
Silhouette Coefficients for k = 60 is 0.2571171853109766
Silhouette Coefficients for k = 61 is 0.26163938132644377
Silhouette Coefficients for k = 62 is 0.2629753320981927
Silhouette Coefficients for k = 63 is 0.2741419714954385
Silhouette Coefficients for k = 64 is 0.2477539128633301
Silhouette Coefficients for k = 65 is 0.25970251889750057
```

Silhouette Coefficients for k = 66 is 0.2629300559488024
Silhouette Coefficients for k = 67 is 0.24374453163267248
Silhouette Coefficients for k = 68 is 0.23894045172887884
Silhouette Coefficients for k = 69 is 0.24971743859990492
Silhouette Coefficients for k = 70 is 0.25293073479675926
Silhouette Coefficients for k = 71 is 0.2487632709412719
Silhouette Coefficients for k = 72 is 0.241182968754236
Silhouette Coefficients for k = 73 is 0.24818484334508403
Silhouette Coefficients for k = 74 is 0.2583519925336459
Silhouette Coefficients for k = 75 is 0.23665148024859178
Silhouette Coefficients for k = 76 is 0.2340966733686849
Silhouette Coefficients for k = 77 is 0.24761872841630517
Silhouette Coefficients for k = 78 is 0.23404215765716563
Silhouette Coefficients for k = 79 is 0.2413720064076678
Silhouette Coefficients for k = 80 is 0.23310457835841916
Silhouette Coefficients for k = 81 is 0.23736005330701224
Silhouette Coefficients for k = 82 is 0.23298958551630003
Silhouette Coefficients for k = 83 is 0.2152387947919705
Silhouette Coefficients for k = 84 is 0.217871478498492
Silhouette Coefficients for k = 85 is 0.2388779014551237
Silhouette Coefficients for k = 86 is 0.2330361694266244
Silhouette Coefficients for k = 87 is 0.2301163667614603
Silhouette Coefficients for k = 88 is 0.2141171974899405
Silhouette Coefficients for k = 89 is 0.2178181858860691
Silhouette Coefficients for k = 90 is 0.2231615322480649
Silhouette Coefficients for k = 91 is 0.22163770144158051
Silhouette Coefficients for k = 92 is 0.22423896734716636
Silhouette Coefficients for k = 93 is 0.2148859267761434
Silhouette Coefficients for k = 94 is 0.19992733403675297
Silhouette Coefficients for k = 95 is 0.20713149203979933
Silhouette Coefficients for k = 96 is 0.20372695450294365
Silhouette Coefficients for k = 97 is 0.20135017173650777
Silhouette Coefficients for k = 98 is 0.20503276320479116
Silhouette Coefficients for k = 99 is 0.19836687356426944
Silhouette Coefficients for k = 100 is 0.2006232061195443
Silhouette Coefficients for k = 101 is 0.19407472077209784
Silhouette Coefficients for k = 102 is 0.1965189287808751
Silhouette Coefficients for k = 103 is 0.18341397729986306
Silhouette Coefficients for k = 104 is 0.178523974196948
Silhouette Coefficients for k = 105 is 0.18580581870868107
Silhouette Coefficients for k = 106 is 0.1867293399708967
Silhouette Coefficients for k = 107 is 0.16939396779310173
Silhouette Coefficients for k = 108 is 0.17086856052948704
Silhouette Coefficients for k = 109 is 0.16441177027581105
Silhouette Coefficients for k = 110 is 0.16877256169254476
Silhouette Coefficients for k = 111 is 0.1637590018380971
Silhouette Coefficients for k = 112 is 0.1694779459766526
Silhouette Coefficients for k = 113 is 0.1541896531304419
Silhouette Coefficients for k = 114 is 0.15055356543909515
Silhouette Coefficients for k = 115 is 0.15326151649622335
Silhouette Coefficients for k = 116 is 0.14654806418579605
Silhouette Coefficients for k = 117 is 0.14402579191579215
Silhouette Coefficients for k = 118 is 0.1428783433959318
Silhouette Coefficients for k = 119 is 0.1407695002182544
Silhouette Coefficients for k = 120 is 0.13521807530277116
Silhouette Coefficients for k = 121 is 0.12873954024932563
Silhouette Coefficients for k = 122 is 0.1246688346421665
Silhouette Coefficients for k = 123 is 0.12405046223388766
Silhouette Coefficients for k = 124 is 0.11997079653448096
Silhouette Coefficients for k = 125 is 0.11570255234456397
Silhouette Coefficients for k = 126 is 0.10839387246562704
Silhouette Coefficients for k = 127 is 0.11177721815086712
Silhouette Coefficients for k = 128 is 0.10497619557238715
Silhouette Coefficients for k = 129 is 0.09897661618697981

```
Silhouette Coefficients for k = 130 is 0.09502592127469935
Silhouette Coefficients for k = 131 is 0.09512266973169638
Silhouette Coefficients for k = 132 is 0.08913655825096725
Silhouette Coefficients for k = 133 is 0.07884476448411151
Silhouette Coefficients for k = 134 is 0.0760054635810227
Silhouette Coefficients for k = 135 is 0.07373493828508416
Silhouette Coefficients for k = 136 is 0.07251158215793115
Silhouette Coefficients for k = 137 is 0.06969391728586213
Silhouette Coefficients for k = 138 is 0.06693249353671078
Silhouette Coefficients for k = 139 is 0.0636528851327739
Silhouette Coefficients for k = 140 is 0.058017555388635116
Silhouette Coefficients for k = 141 is 0.05151464781990076
Silhouette Coefficients for k = 142 is 0.0458767836644781
Silhouette Coefficients for k = 143 is 0.044135274390054706
Silhouette Coefficients for k = 144 is 0.04291191826290374
Silhouette Coefficients for k = 145 is 0.038141631932074344
Silhouette Coefficients for k = 146 is 0.030146745172696734
Silhouette Coefficients for k = 147 is 0.026241502255184037
Silhouette Coefficients for k = 148 is 0.022336259337667152
Silhouette Coefficients for k = 149 is 0.013333333333333334
```

Plot the results as Silhouette Coefficient x number of clusters

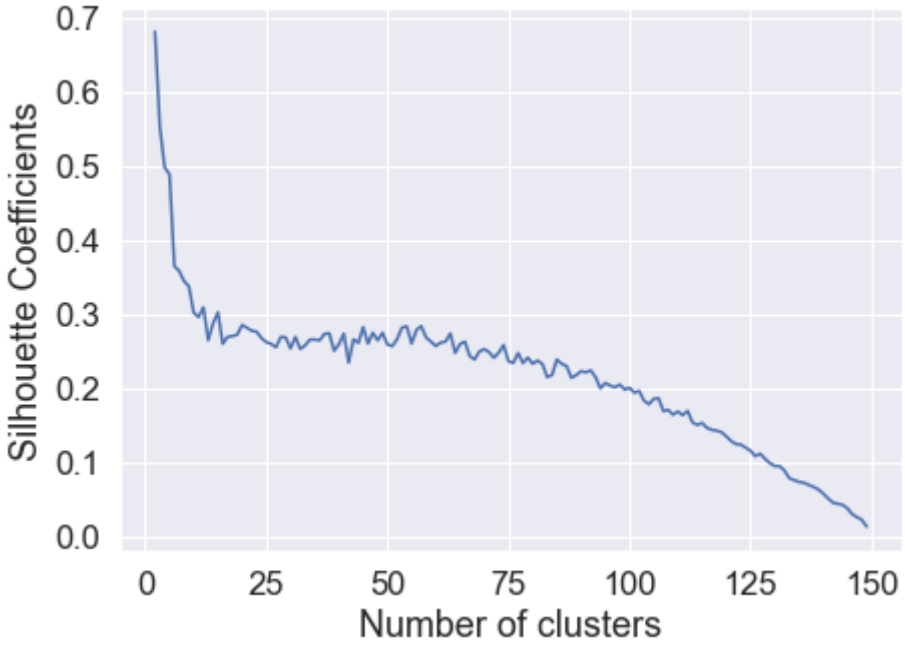
In [ ]:
```
# Plot the results as Silhouette Coefficient x number of clusters

f, ax = plt.subplots(figsize=(7, 5))
ax.plot(range(min_k, len(X)), silhouette_avgs)

plt.xlabel("Number of clusters")
plt.ylabel("Silhouette Coefficients")

#---the optimal k is the one with the highest average silhouette---
Optimal_K = silhouette_avgs.index(max(silhouette_avgs)) + min_k
print("Optimal K is ", Optimal_K)
```

```
Optimal K is  2
```



Conclusions

As shown and visialized above, we used the unlabeled Iris dataset to train and test our k-Means model. We started with three clusters (K=3), and calculated the Silhouette Coefficient to be **~0.553**.

Next, to find the optimal number of clusters for our model, we iteratively built, trained and calculated the Silhouette Coefficient for the same model while varying the K values from 2 up to the number of target datapoints in the dataset (149). We found the model to perform best with this dataset with two clusters (K=2) as it gave the best Silhouette Coefficient of **~0.68**. The plot above shows the change in the Silhouette Coefficient as we varied the number of clusters.

Overall, the algorithim we followed here can be considered the best approach to build, train, validated, and improve a K-Means clustering model. Although, skipping the step where we built the model using the number of classes in the dataset as the value of K, and runing an iterative test directly to find the best K, would have been better approach. This is due to the nature of the K-Means model and the need to define the number of clusters beforehand (K).